

Evaluation of MPEG-4 Video Streaming over UMTS/WCDMA Dedicated Channels

Anthony Lo[†], Geert Heijenk[‡] and Ignas Niemegeers[†]

[†]*Delft University of Technology*
P O Box 5031, 2600 GA Delft
The Netherlands
[A.Lo, I.Niemegeers]@ewi.tudelft.nl

[‡]*University of Twente*
P O Box 217, 7500 AE Enschede
The Netherlands
Geert.Heijenk@utwente.nl

Abstract

Universal Mobile Telecommunications System (UMTS) is a third-generation mobile communication system that supports wireless wideband multimedia communications. This paper investigates the performance of streaming MPEG-4 video over UMTS dedicated channels with varying channel conditions. In order to evaluate the performance, simulation models for UMTS and MPEG-4 were developed. Simulation results show that UMTS Radio Control Link (RLC) protocol operating in the acknowledged mode outperforms the unacknowledged mode. The latter mode provides timely delivery but no error recovery. The acknowledged mode can deliver acceptable video quality for RLC block error rates up to 40% utilizing a pre-decoder buffer. We also discover that RLC block size can affect the video quality with a larger RLC block size attaining better quality. However, the optimum RLC block size is dependent on the encoded video frame size and the channel bit rate. Therefore, the RLC block size can be tuned to a proper value given the encoded video frame size and channel bit rate.

1. Introduction

Universal Mobile Telecommunications System (UMTS) [1]-[2] is a third-generation mobile communication system where the radio interface is based on code division multiple access, known as Wideband Code Division Multiple Access (WCDMA). To date, UMTS networks are in commercial service and many other UMTS networks around the world are in either pre-commercial or deployment phase. UMTS is among the first third-generation mobile system to offer wireless wideband multimedia communications over the Internet protocol [3]. As such, it allows

mobile Internet users to access a variety of multimedia contents available on the Internet in a seamless fashion (i.e., always on) at data rates up to 384 kb/s in wide-area coverage and high user mobility.

Video streaming is a multimedia service, which is recently gaining popularity and expected to unlock new revenue flows for mobile network operators. Typically, users access online video clips by clicking on a hyperlink using their web browser, which results in the browser opening a video player to play the selected clip. With streaming, the video content need not be completely downloaded, but the client can begin playout the video a few seconds after it begins receiving parts of the content from the streaming server. Since raw video requires high transmission bit rates, video compression is usually employed to achieve transmission efficiency. MPEG-4 [4] is a video coding standard adopted by most mobile networks including UMTS. The compression technology of MPEG-4 produces good quality video at bit rates that are suitable for mobile and wireless transmission.

The paper investigates the video quality achieved by streaming MPEG-4 video over UMTS networks. We employ a simulation-based approach to evaluate the streaming video quality with varying channel conditions.

2. Universal Mobile Telecommunications System (UMTS)

2.1. System Architecture

Fig. 1 shows a simplified architecture of UMTS for IP domain or packet-switched mode [2][5], which consists of one or several User Equipments (UEs), the UMTS Terrestrial Radio Access Network (UTRAN) and the core network. The UTRAN is composed of several Node Bs connected to a Radio Network

Controller (RNC). The core network, which is the backbone of UMTS, comprises the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The SGSNs route packets to and from UTRAN, while GGSNs interface with external IP networks. UE, which is a mobile station, is connected to Node B over the UMTS radio interface.

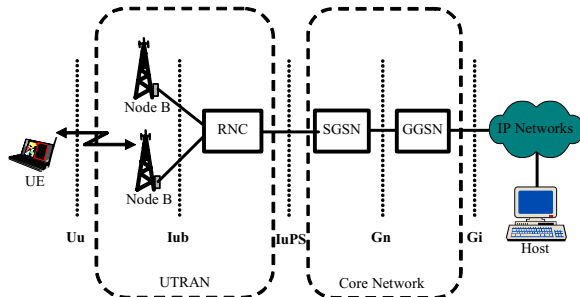


Figure 1. UMTS Reference Architecture

2.2. Protocol Architecture

Fig. 2 depicts the UMTS protocol architecture for the transmission of user data which is generated by IP-based applications. The applications as well as the Internet protocol suite are located at the end-nodes, namely, the UE and a host.

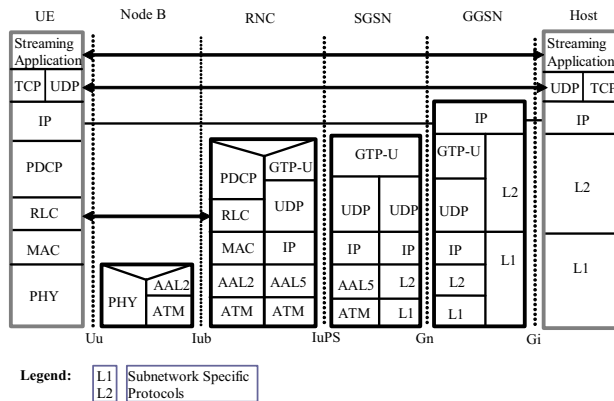


Figure 2. UMTS Protocol Architecture – U-Plane

The Packet Data Convergence Protocol (PDCP) provides header compression functionality. The Radio Link Control (RLC) layer can operate in three different modes: acknowledged, unacknowledged and transparent. The acknowledged mode provides reliable data transfer over the error-prone radio interface. Both the unacknowledged and transparent modes do not guarantee data delivery. The transparent mode is targeted for the UMTS circuit-switched mode in which data are passed through the RLC unchanged. The Medium Access Control (MAC) layer can operate in either dedicated or common mode. In the dedicated mode, dedicated physical channels are allocated and

used exclusively by one user (or UE), whereas in the common mode, users share common physical channels for transmitting and receiving data. The Physical (PHY) layer contains, besides all radio frequency functionality, spreading, and the signal processing including, power control, forward error-correction and interleaving.

3. MPEG-4

MPEG-4 [4] is a video codec standardized by ISO/IEC. The standard defines tools to create, represent and distribute individual audiovisual objects, both natural and synthetic, ranging from arbitrarily shaped objects to sprites, face and body animations.

An MPEG-4 encoder generates three types of frames. Intra-frames (I-frames) contain information from encoding a still image. Predictive frames (P-frames) are encoded from the previous I-frames or P-frames. Bi-directional frames (B-frames) are encoded bi-directionally from the preceding and the following I- and P-frames. B-frames achieve the highest compression, which require the lowest transmission bandwidth, but they also take the longest time to encode. Conversely, I-frames have the lowest compression of the three frame types, but they can be encoded and decoded faster than the other frame types.

4. Simulation Models

In order to analyze the performance of MPEG-4 video streaming over UMTS, network simulations were carried out using *ns-2* [6], which is an event-driven simulator. Several extensions were made to this simulator for modeling UMTS and MPEG-4 video streaming. With the extensions, instances of UMTS nodes, viz., UE, Node B and RNC can be instantiated. The network model used for simulation analysis is illustrated in Fig. 3. Note that, the model is based on the system architecture described in the previous section, see Fig. 1. The basic entities of video streaming service consist of a streaming client and a streaming server. In the simulation model, a UE plays the role of a streaming client and a fixed host is the streaming server located in the Internet.

Video streaming uses a family of transport protocols, namely, User Datagram Protocol (UDP), Real-time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP). UDP is a lower-layer transport protocol while RTP and RTCP are upper-layer transport protocols. UDP is employed because it provides timely delivery of packets. However, UDP does not guarantee packet delivery. RTP runs on top of

UDP, which packetizes and provides in-order delivery of video frames. RTCP is used by the video client to inform video server concerning the received video quality. In the simulation, we assume no interaction between the video client and server. Hence, RTCP is not modeled. Similarly, RTP is also not modeled, but those functions that are needed such packetization, packet sequence numbering and in-order delivery are added to the UDP model of *ns-2*.

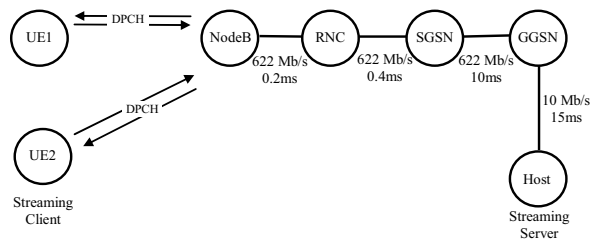


Figure 3. Top-level Simulation Model

Since the primary aim of the simulation was to investigate the impact of the radio interface on the quality of MPEG-4 video streaming, we assume that no packet losses, errors or congestion occur on either the Internet or the UMTS core network. The transfer delay introduced by the Internet and the UMTS core network is constant throughout the entire video streaming duration. Hence, the quality of the streaming video is solely attributed to the radio interface. The links between two nodes are labeled with their bit rate (in bits per second) and delay (in seconds). Each link capacity was chosen so that the radio channel is the connection bottleneck. Hence, the quality of the streaming video is solely attributed to the UMTS radio interface. Consequently, the functionality of SGSN and GGSN was abstracted out and modeled as traditional *ns* nodes since generally they are wired nodes and, in many ways, mimic the behavior of IP router. Currently, no header compression technique is supported in the PDCP layer. In the following subsections, the MPEG-4 and UMTS models are described in detail.

4.1. Trace-based Video Traffic Model

A generic and independent video source model for network performance simulation is not feasible because the encoded video traffic is dependent on the video contents, the encoding standard and the encoder settings. Hence, a trace-based approach is employed because it is widely applied to performance analysis of video traffic using network simulators [7]-[9]. The trace-based approach is illustrated in Fig. 4, which comprises several components: an *MPEG-4 encoder*, a *post-simulation processing* program, and an *MPEG-4*

decoder. Both the MPEG-4 encoder and decoder were written in the C programming language using an open-source MPEG-4 video codec library [10]. The post-simulation processing program was also written in C.

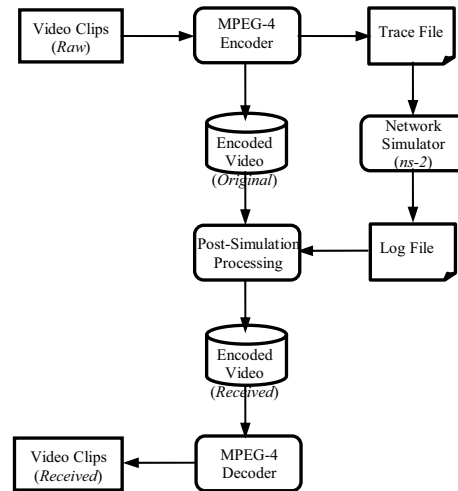


Figure 4. Streaming Video Model

A video clip is fed to an MPEG-4 encoder which generates video traces and an encoded video stream. The generated video traces are captured in a *trace* file, which contains a row of four entries for each video frame: *departure time* (in seconds) is the time when a frame is generated at the streaming server, *frame index* which starts at 0 and incremented by one for each new video frame, *frame type* (e.g., I, P or B), and *frame size* in bytes. The trace file is then fed to the streaming server in the *ns-2* environment to generate video streams over UMTS. The effect of streaming video over UMTS is captured in a *log* file which is generated by *ns-2*. The log file contains information used by the *post-simulation processing* program to impart the quality of the transmission on each encoded video frame received at the streaming client. The log file contains a row of six entries for each video frame. The first four entries include the same information as in the trace file. The arrival time of each video frame is recorded in the fifth column and the last column indicates the frame status whether the frame is error-free or not. Once the encoded video frame has been processed, it can be fed to an MPEG-4 decoder. The decoded video can be used for objective quality assessment and played back to observe the quality of wireless transmission.

4.2. UMTS Network Models

4.2.1. RLC Model. The RLC model supports both *acknowledged* and *unacknowledged* operation modes.

The unacknowledged mode (UM) provides unreliable but timely delivery of RLC blocks. That means, no error recovery is performed even though erroneous RLC blocks are detected. An RLC block consists of a header and a payload which carries higher layer data. An erroneous block is discarded by RLC, which would result in an incomplete IP packet. An incomplete IP packet is discarded by RLC and not delivered to higher layers as soon as one of the RLC blocks composing the IP packet is erroneous.

On the other hand, the acknowledged mode (AM) guarantees delivery by retransmitting erroneous RLC blocks at the expense of transfer delay. The retransmission strategy adopted by the acknowledged mode is the Selective-Repeat ARQ (Automatic Repeat reQuest) scheme. In the simulation, the ARQ was configured to operate in a persistent manner. With Selective-Repeat ARQ, the only RLC blocks retransmitted are those that receive a negative acknowledgement. A status message is used by the receiver for notifying loss or corruption of an RLC block. The status message is in bitmap format. That is, bit_j indicates whether the j th RLC block has been correctly received or not. The frequency of sending status messages is not specified in the standard [11]. However, several mechanisms are defined, which can trigger a status message. Either the sender or the receiver can trigger the status message. Table 1 and Table 2 list the triggering mechanisms for sender and receiver, respectively. It is important to note that not all the triggering mechanisms are needed for the Selective-Repeat ARQ to operate. However, a combination of triggering mechanisms, which deliver optimum performance, is sought. The advantage of receiver-initiated mechanisms is that the receiver has direct information about missing blocks. For the sender-initiated mechanisms, the sender has first to request a status message by enabling the POLL flag in the RLC header and wait for a reply, which has longer turn around time. Therefore, receiver-initiated mechanisms are preferred. Nevertheless, sender-initiated mechanisms are required to prevent deadlocks and stall conditions. Periodic mechanisms might be more robust compared to others but may result in too frequent status message. In addition, a timer is required at the sender and receiver for proper operation of the triggering mechanisms. At the sender, the timer is called *poll* timer, which is started when a request for status messages is sent to the receiver. If the status message from the receiver does not arrive before the timer expires, the sender repeats the same procedure again. The receiver is equipped with a timer called *status prohibit* timer, which controls the time interval between status messages if triggered consecutively. If

the interval is too short, then bandwidth is wasted. On the other hand, if the interval is too long, bandwidth is preserved, but delay increases. The selected triggering mechanisms for the RLC model are labeled by the rows in dark grey.

Table 1. Sender-Initiated Mechanisms

Trigger	Explanation
Last Block in buffer or retransmission buffer	status report is requested by enabling the poll flag in the RLC header
Every m blocks	poll flag is enabled for every m blocks
Every n service data units	poll flag is enabled for every n service data units
Utilization of Send Window	poll flag is enabled when the Send Window is $x\%$ full
Periodic Poll	poll is generated periodically based on a timer

Table 2. Receiver-Initiated Mechanisms

Trigger	Explanation
Detection of missing blocks	Status message is generated once a gap is detected in the RLC sequence number
Periodic status	Status message is sent periodically based on a timer
Estimated block Counter	Status message is generated if not all the retransmitted blocks are received within an estimated period

4.2.2. MAC Model. The MAC model implemented the dedicated mode. It requests a number of blocks buffered at the RLC layer, which are ready for transmission, and submits to the PHY layer as transport blocks. In this case, each transport block corresponds to an RLC block since no MAC header is required in the dedicated mode as depicted in Fig. 5. The frequency at which the PHY layer can accept transport blocks from MAC is defined by the Transmission Time Interval (TTI). In the UMTS standard, the values of TTI are 10 ms, 20 ms, 40 ms and 80 ms.

4.2.3. PHY Model. The PHY model is responsible for transmitting transport blocks over the physical channels. For the MAC dedicated mode, the transport blocks are sent over the Dedicated Physical Channel (DPCH) which maintains a fixed bit rate for the duration of the connection. The channel bit rates and the TTI associated with the channel considered in the simulation are shown in Table 3. Note that the bit rates exclude the RLC headers. The other PHY layer functionality is not implemented in the model. Since the PHY layer passes the transport block to the MAC layer together with the error indication from the Cyclic Redundancy Check (CRC), the output of the PHY layer can be characterized by the overall probability of transport block error – referred to as Transport Block Error Rate (TBLER) in this paper. Thus, an error model based on uniform distribution of transport block errors, was used in the simulation. It is valid to assume

that the erroneous transport blocks perceived by the RLC is independent and uniformly distributed as a result of interleaving, forward error-correction and fast power control mechanisms provided by the PHY layer [12]. The TBLER, in the range from 0 to 50%, was considered in the simulation.

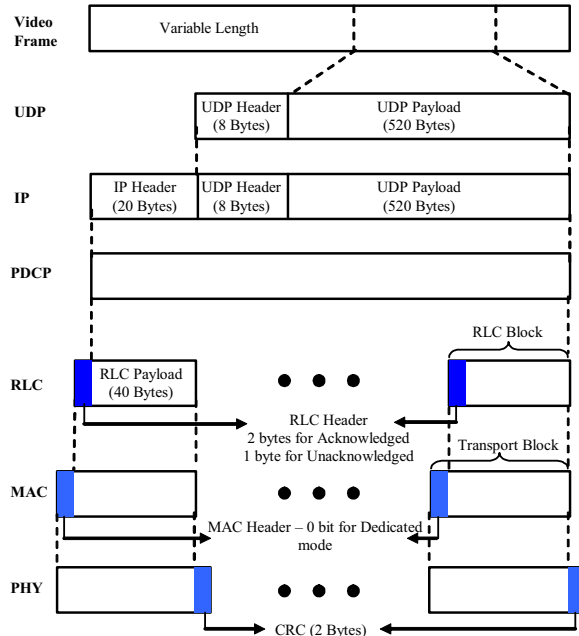


Figure 5. IP Packet Data Transfer

The transmission of video over the UMTS network is illustrated in Fig. 5. Each encoded video frame is carried by UDP over the IP protocol. If the video frame size is larger than the UDP segment size of 520 bytes then it is fragmented. The IP packets are encapsulated into PDCP packets. The RLC entity receives a PDCP packet which comprises an IP packet of 548 bytes. This PDCP packet is segmented into multiple RLC blocks of fixed size. Each of these blocks fits into a transport block in which a CRC is attached at the PHY layer. In the simulation, the RLC header for the acknowledged and unacknowledged modes is 2 bytes and 1 byte, respectively, while the payload size is 40 bytes for both modes. For this RLC payload size and, for instance, a channel bit rate of 384 kb/s, twelve transport blocks are transmitted within one TTI of 10 ms. The simulation parameters are summarized in Table 3.

5. Simulation Results

In order to assess the impact of UMTS radio interface on the video quality, two performance metrics are used, namely *Peak Signal to Noise Ratio* (PSNR) and *end-to-end video frame delay*. PSNR is an

objective indicator of video quality. For a video frame of size, $N \times M$ pixels, the PSNR is computed by the formula [13]

$$PSNR = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (f(i,j) - \hat{f}(i,j))^2}} \text{ dB}$$

where $f(i, j)$ and $\hat{f}(i, j)$ represent the value of pixel (i, j) in the raw and decoded video source frames, respectively. The quality of a video frame with a PSNR value of less than 25 dB is generally considered bad and unacceptable by viewers. The end-to-end video frame delay is the time when a video frame is generated at the streaming server until it is received by the streaming client.

Table 3. Simulation Parameters

App	MPEG-4				
UDP	Maximum Segment Size (Bytes)	520			
	UDP Header Size (Bytes)	8			
IP	IP Header Size (Bytes)	20			
	IP Packet Loss Rate in the Internet	0%			
PDCP	UDP/IP Header compression	No			
RLC	RLC Mode	Acknowledged	Unacknowledged		
	Window Size (Blocks)	4096	Not Applicable		
	Payload Size (Bits)	[320, 640]	[320, 640]		
	RLC Header (Bits)	16	8		
MAC	MAC Header (Bits)	0			
	MAC Multiplexing	Not required for DPCH			
PHY	Physical Channel Type	DPCH			
		Uplink		Downlink	
	Bit Rate (kb/s)	TTI (ms)	Bit Rate (kb/s)	TTI (ms)	
	64	20	384	10	
	Transport Block Size (Bits)	336			
Transport BLER	0 – 50%				
Error Model	Uniform Distribution				

Two video clips with distinct characteristics were selected for the simulations. The video clips are depicted in Figs. 6 and 7. The first clip is called “Music” which is 750 frames long encoded at 25 frames/s. The second one is “Container” comprises 300 frames long encoded at 30 frames/s. The frame size of both video clips is $N=176 \times M=144$ pixels, which is also known as Quarter Common Intermediate Format (QCIF). The QCIF format is used because it is a typical format for streaming video over mobile networks. Table 4 summarizes the characteristics of the video clips. Note that the source bit rate of both video clips is less than the downlink channel bit rate. Note that, all the results presented in the next

subsection were obtained over 25 simulation runs using random seeds with a 98% confidence interval.



Figure 6. Music



Figure 7. Container

Table 4. Characteristics of Video Clips

Video Clip	Music	Container
Average Video Bit Rate	343 kb/s	226 kb/s
Average Video Frame Size	1,712 bytes	938 bytes
Temporal Information	High	Low

5.1. Effect of RLC Mode

Fig. 8 plots the average decoded PSNR for the two video clips at the client as a function of TBLER for RLC AM and UM modes. Under ideal radio channel, both the RLC modes obtained the same average PSNR, which is 40.01 dB and 40.69 dB for “Music” and “Container”, respectively.

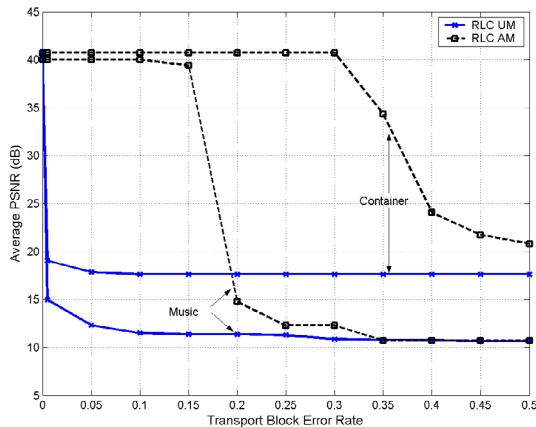


Figure 8. Average PSNR as a function of Transport Block Error Rate

As TBLER increases, the average PSNR for AM is superior to UM for both video clips. In the case of UM, the quality for the two video clips drops below the acceptable threshold (25 dB) once the TBLER exceeded 0.5% TBLER. As mentioned, RLC UM does not recover erroneous RLC blocks. Instead, the erroneous RLC blocks are discarded including the contents of the entire IP packets. As a result, the decoder at the streaming client has to decode video frames with missing data parts. The video quality is

very sensitive to missing data. The poor video quality achieved by UM is clearly evidenced when the TBLER is mapped to video frame error rate which is defined as a video frame with missing data parts. The video frame error rate is shown in Fig. 11. For TBLER as low as 5%, the video frame error rate is approximately 60% and 90% for “Container” and “Music”, respectively. Such a high video frame error rate can cause any video decoder to fail. For a given TBLER, the video frame error rate of “Music” is higher because it contains larger encoded video frame size. The video quality for AM drops below the acceptable threshold only when the TBLER exceeded 17% and 40% for “Music” and “Container”, respectively using a fixed pre-decoder buffering time of $200 \times TTI$ (i.e., 2s). The TBLER cut-off point where the video quality drops below the acceptable threshold is proportional to the buffering time. In other words, a longer buffering time can handle higher TBLER.

In addition, the TBLER cut-off point varies with the video content. The “Container” video clip can tolerate higher TBLER because it has smaller video frame size than “Music”. This leads to shorter end-to-end video frame delay for the same TBLER. This is clearly evidenced in Fig. 9 and Fig. 10. The figures depict the Cumulative Density Function (CDF) of the end-to-end video frame delay for “Music” and “Container”, respectively. For instance, for TBLER at 30%, the maximum end-to-end video frame delay for “Music” is approximately increased by 10 times as compared with “Container”. The end-to-end video frame delay has a drastic impact on the pre-decoder buffering time at the streaming client. A larger end-to-end video frame delay causes the pre-decoder buffer to underflow at the time of decoding a video frame, which leads to video quality degradation. As observed in Fig. 8 for RLC AM, the video quality for both clips is rapidly deteriorating once the TBLER cut-off point is exceeded due to the increase in end-to-end video frame delay which causes pre-decoder buffer to underflow. The end-to-end video frame delay increases significantly for larger TBLERs because the probability of some RLC blocks that need to be retransmitted is also higher. With the number of retransmissions rises, the delay increases by multiples of TTIs. Additionally, the in-sequence delivery of RLC AM delays all correctly received IP packets until all previous IP packets have been correctly received. This delays buffered IP packets at RLC according to the slowest IP packet.

A large pre-decoder buffer allows more erroneous RLC blocks to be recovered, which improves video

quality at the expense of longer start-up time experienced by the viewer and extra storage capacity at the streaming client. For a 384 kb/s transport channel and 2s buffering time, a buffer size of at least 96 kbytes is required.

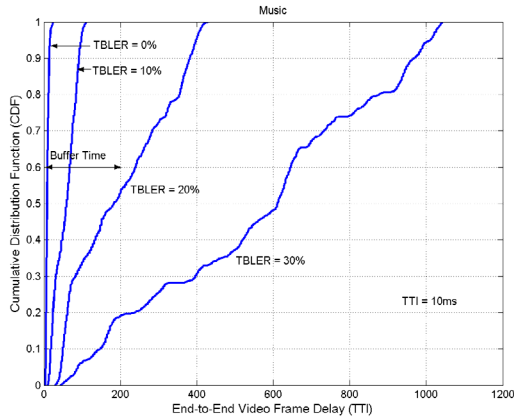


Figure 9. End-to-End Video Frame Delay Cumulative Distribution Function for RLC AM

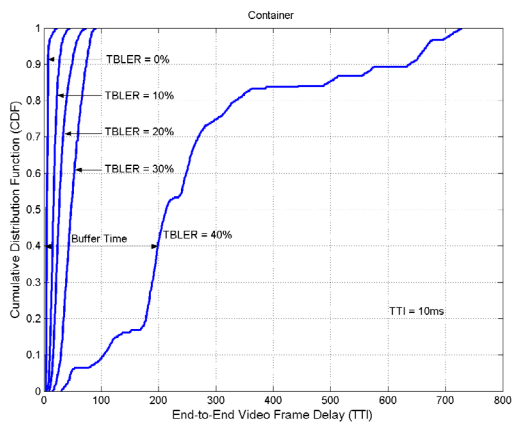


Figure 10. End-to-End Video Frame Delay Cumulative Distribution Function for RLC AM

5.2. Effect of RLC Block Size

In this subsection, we investigate the impact of RLC block size on the video quality. The RLC block size was doubled from 40 bytes to 80 bytes and the same set of simulation was repeated. Fig. 12 shows the average PSNR for the two video clips as a function of TBLER. Similarly, the average PSNR for AM outperforms UM for both video clips. In addition, the plot for “Container” indicates performance gain compared with the case of 320 bits block size. That is, the TBLER cut-off point is increased by 10%. In order to determine the reason for the performance gain, the end-to-end video frame delay of “Container” was examined. Fig. 13 plots the CDF of the end-to-end video frame delay for different TBLERs. As observed

in the figure, all the plots depict shorter end-to-end video frame delay as compared with Fig. 10. For “Music”, a similar RLC block size failed to reduce the end-to-end video frame delay because its encoded video frame size is larger than “Container”.

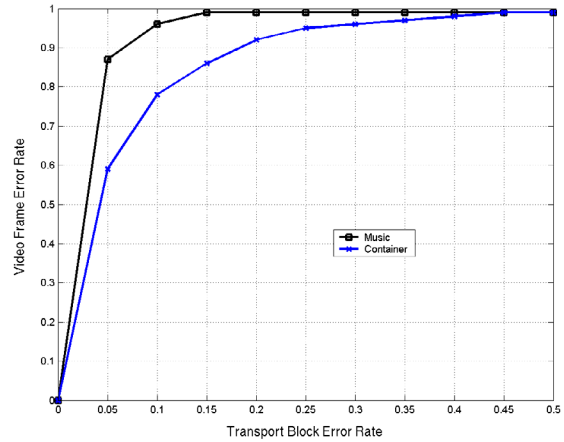


Figure 11. Video Frame Error Rate for “Music” and “Container” for RLC UM

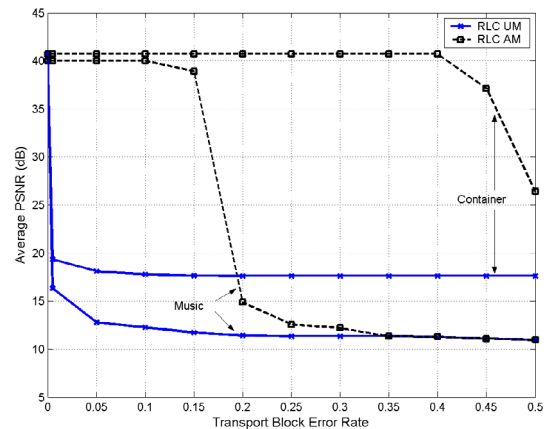


Figure 12. Average PSNR as a function of Transport Block Error

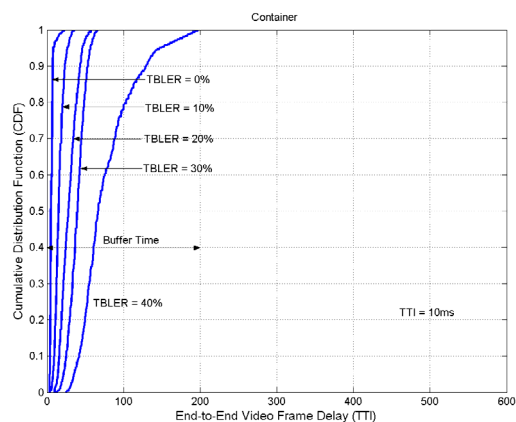


Figure 13. End-to-End Video Frame Delay Cumulative Distribution Function for RLC AM

6. Conclusion

The paper has evaluated the performance of streaming MPEG-4 video over UMTS dedicated channels with varying channel conditions. In order to evaluate the performance, a trace-based video traffic model was implemented because it allows both objective and subjective video quality assessment. The objective performance metrics are PSNR and end-to-end video frame delay. Two video clips with distinct characteristics were considered. Simulation results show that the video quality achieved using UMTS RLC acknowledged mode is superior to unacknowledged mode. The performance of unacknowledged mode is worse because it does not provide error recovery, and the video decoder is sensitive to erroneous data and error propagation. The acknowledged mode can tolerate transport block error rates up to 40% before the video quality deteriorates due to the increase in end-to-end video frame delay which causes the pre-decoder buffer to underflow. Furthermore, for acknowledged mode, we discover that a larger RLC block size can reduce end-to-end video frame delay which in turn gives performance gain. In summary, the RLC acknowledged mode is recommended for streaming video over UMTS, and the RLC block size can be tuned according to the encoded video frame size and channel bit rate.

References

- [1] 3GPP, available from <http://www.3gpp.org>.
- [2] H. Kaaranen, S. Naghian, L. Laitinen, A. Ahtianen, and V. Niemi, *UMTS Networks: Architecture, Mobility and Services*, New York: John-Wiley & Sons, 2001.
- [3] W. Stevens, *TCP/IP Illustrated*, Vol 1, Reading, MA: Addison Wesley, 1994.
- [4] R. Koenen, "Profiles and Levels in MPEG-4: Approach and Overview", *Image Communication Journal*, 15, 2000.
- [5] 3GPP, "Network architecture (Release 6)", TS 23.002, 2004.
- [6] K. Fall and K. Varadhan, "The ns Manual", <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [7] F. H. P. Fitzek, P. Seeling, and M. Reisslein, *Using Network Simulators with Video Traces* (Tech. Rep.). Tempe, AZ: Arizona State University, Department of Electrical Engineering, 2003. <http://trace.eas.asu.edu/pub.html>
- [8] F. H. P. Fitzek and M. Reisslein, "MPEG-4 and H.263 Video Traces for Network Performance Evaluation", *IEEE Network*, 15, 2001.
- [9] J. Klaue, B. Rathke and A. Wolisz, "EvalVid – A Framework for Video Transmission and Quality Evaluation", Proceedings of the 13th International Conference on Modeling, Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, 2003.
- [10] XviD, MPEG-4 Video Codec Library, 2002. <http://www.xvid.org>
- [11] 3GPP, "Radio Link control (RLC) Protocol Specification (Release 6)", TS 25.322, 2004.
- [12] M. Meyer, J. Sachs and M. Holzke, "Performance Evaluation of TCP Proxy in WCDMA Networks", *IEEE Wireless Communications*, vol. 10, no. 5, 2003.
- [13] L. Hanzo, P. J. Cherriman and J. Streit, *Wireless Video Communications*, Digital & Mobile Communications, IEEE Press, Piscataway, 2001.